AF
2700
#

PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF APPEALS

EX PARTE HON WAH CHIN

Application for Patent

Filed September 30, 1998

Serial No. 09/164,388

FOR:

REDUCING CPU OVERHEAD IN THE FORWARDING PROCESS

APPEAL BRIEF

CERTIFICATE OF MAILING

I hereby certify that this correspondence is being deposited with the United States Postal Service as First Class Mail in an envelope addressed to: Commissioner of Patents and Trademarks, P.O. Box 1450, Alexandria, VA 22313-1450 on March 30, 2004.

Signed: Tiffany Bell
Tiffany Bell

BEYER WEAVER & THOMAS, LLP
Attorneys for Applicant

TABLE OF CONTENTS

I. REAL PARTY IN INTEREST

The real party in interest is Cisco Technology, Inc., the assignee of the present application.

II. RELATED APPEALS AND INTERFERENCES

The undersigned is not aware of any related appeals and/or interferences.

III. STATUS OF THE CLAIMS

There have been a total of 54 claims presented throughout the prosecution of this application. Claims 1-43 and 45-54 are pending and stand rejected in this application and are subject to this appeal. Claim 44 has been cancelled.

Claims 1-43 and 45-54 stand rejected under 35 USC §103(a).

IV. STATUS OF AMENDMENTS

On October 30, 2003, the Examiner entered a final rejection. The applicants previously filed Amendment F on July 14, 2003, which was **entered**.

V. SUMMARY OF INVENTIONS

The present invention relates generally to methods and apparatus for reducing CPU overhead in the forwarding process. More particularly, the present invention relates to forwarding packets in a network device (e.g., router) such that per packet CPU involvement typically required when moving a packet from an inbound interface to an outbound interface is reduced or eliminated. This is accomplished, in part, by supporting the simultaneous transfer of multiple packets (e.g., queue of packets).

As described in Applicant's specification, the traditional router has several potentially undesirable characteristics. First, an inbound port of a router typically has a single inbound queue associated therewith. Second, when a packet is forwarded, a single packet or entry in the inbound queue is transferred by an inbound controller to an outbound controller. Third, when the packet is received by the outbound controller, information associated with a single packet is stored in an entry in an outbound queue. Thus, a substantial amount of CPU overhead is consumed during the forwarding process.

Each of the independent claims provides at least one of the following advantages or limitations over the prior art. First, a plurality of inbound queues is provided for a single inbound port. An inbound packet is therefore classified in one of the plurality of inbound queues to enable the inbound packet to be stored in the appropriate queue. Second, one of the plurality of inbound queues is transferred to an outbound controller and/or outbound queue capable of storing (or identifying) a multiplicity of inbound queues. In other words, a queue of packets rather than a single packet is transferred to the outbound controller and/or an associated outbound queue such that a reference (e.g., pointer) to the queue of packets is stored in a single entry in the outbound queue. In other words, a reference to each of the multiplicity of inbound queues is stored in a different entry in the outbound queue. Third, in some embodiments of the invention, it is possible to encrypt an inbound queue prior to transmission by an outbound controller. One or more of the above-described limitations are present in each of the claims. In this manner, CPU overhead in the forwarding process is reduced.

VI. ISSUES

The issues which applicant believes to be most pertinent to the present appeal include:

(a) Whether the combination of U.S. Patent No. 6,487,212 issued to Erimli (Erimli) and U.S. Patent No. 5,392,401 issued to Barucchi (Barucchi) reasonably suggests an **inbound controller** or inbound controller method wherein the inbound controller determines when one of a plurality of inbound queues is ready to be moved to an entry in an outbound queue. (Claims 1-9, 20-27).

(b) Whether any combination of Erimli, Barucchi, and U.S. Patent No. 5,177,480 issued to Clark (Clark) reasonably suggests an **inbound controller** or inbound controller method wherein the inbound controller determines when one of a plurality of inbound queues is ready to be moved to an entry in an outbound queue. (Claims 45-48, 52)

(c) Whether the combination of Erimli and Barucchi reasonably suggests an **outbound controller** or outbound controller method wherein the outbound controller transfers an inbound queue storing a plurality of packets to one of a plurality of entries in an outbound queue. (Claims 10-18, 28-36).

(d) Whether any combination of Erimli, Barucchi and Clark reasonably suggests an **outbound controller** or outbound controller method wherein the outbound controller transfers an inbound queue storing a plurality of packets to one of a plurality of entries in an outbound queue. (Claim 51).

(e) Whether the combination of Erimli and Barucchi reasonably suggests a **router** or router method in which a packet that is received is classified in one of a plurality of inbound queues associated with an inbound port and wherein one of the plurality of inbound queues storing a plurality of packets is transferred to one of a plurality of outbound queues. (Claim 19)

(f) Whether any combination of Erimli, Barucchi and Clark reasonably suggests a **router** or router method in which a packet that is received is classified in one of a plurality of inbound queues associated with an inbound port and wherein one of the plurality of inbound queues storing a plurality of packets is transferred to one of a plurality of outbound queues. (Claims 53, 54)

(g) Whether the combination of Erimli and Barucchi reasonably suggests a **router** in which a plurality of inbound queues are associated with one of a plurality of inbound ports, a plurality of outbound queues are associated with a plurality of outbound ports, and wherein each of the outbound queues is capable of storing a plurality of inbound queues such that a reference to each of the plurality of inbound queues is stored in a different one of a plurality of entries in the outbound queue. (Claim 37-42)

(h) Whether any combination of Erimli, Barucchi, and U.S. Patent No. 5,177,480 issued to Clark (Clark) reasonably suggests an **encryption system** in which an encryption box is adapted for encrypting one of a plurality of inbound queues received by an outbound controller and wherein the outbound controller includes an outbound classifier adapted for classifying the encrypted inbound queue in one of a plurality of outbound queues and storing a reference to the encrypted inbound queue in a single entry in the one of the plurality of outbound queues. (Claim 43, 49-50)

VII. GROUPING OF CLAIMS

With regard to issue (a), the rejected claims do not stand or fall together, claims 1 and 20 will be argued as a group and claims 2-9 and 21-27 will be argued independently.

With regard to issue (b), the rejected claims do not stand or fall together, and each of the claims will be argued independently.

With regard to issue (c), the rejected claims do not stand or fall together, and each of the claims will be argued independently.

With regard to issue (f), claims 53 and 54 will be argued as a group.

With regard to issue (g), the rejected claims do not stand or fall together, and each of the claims will be argued independently.

With regard to issue (h), the rejected claims do not stand or fall together, and each of the claims will be argued independently.

VIII. ARGUMENT

(a) The combination of Erimli and Barucchi neither discloses nor suggests the invention of claims 1-9, 20-27

The Examiner has rejected claims 1-9 and 20-27 under 35 U.S.C. §103(a) as being unpatentable over U.S. Patent No. 6,487,212 (Erimli) in view of U.S. Patent No. 5,392,401 (Barucchi). Independent claims 1 and 20 relate to an **inbound controller** or inbound controller method wherein the inbound controller determines when an inbound queue is ready to be moved to an entry in an outbound queue. In this manner, per-packet processing is substantially reduced. More specifically, pending claim 1 recites:

1. A method for providing an inbound controller for a router, the router having an inbound port and an outbound port, a memory, and a CPU, the inbound controller being adapted for receiving an inbound packet at the inbound port, the method comprising:

providing a plurality of inbound queues for the inbound port;

receiving an inbound packet at the inbound port;

classifying the inbound packet in a selected one of the plurality of inbound queues according to packet sorting criteria;

storing the inbound packet in the selected one of the plurality of inbound queues; and

determining when one of the plurality of inbound queues storing a plurality of packets is ready to be moved to an entry in an outbound queue associated with the outbound port, **the outbound queue being capable of storing a reference to a multiplicity of inbound queues such that a reference to each of the multiplicity of inbound queues is separately stored in a different one of a plurality of entries in the outbound queue, each of the**

multiplicity of inbound queues storing a plurality of packets to be separately transmitted. (Emphasis added).

Pending claim 20 recites:

20. An inbound controller for a router, the router having an inbound port and an outbound port, a memory, and a CPU, the inbound controller being adapted for receiving an inbound packet at the inbound port, comprising:

a packet receiving module coupled to the inbound port, the packet receiving module being adapted for receiving an inbound packet;

wherein the memory has stored therein:

a plurality of inbound queues for the inbound port;

a classifier adapted for classifying the inbound packet in a selected one of the plurality of inbound queues according to packet sorting criteria;

a packet storing module coupled to the classifier, the packet storing module being adapted for storing the inbound packet in the selected one of the plurality of inbound queues; and

a module adapted for **determining when one of the plurality of inbound queues is ready to be moved to an entry in an outbound queue associated with the outbound port, the outbound queue being capable of storing a multiplicity of inbound queues, a reference to each of the multiplicity of inbound queues being stored in a different one of a plurality of entries in the outbound queue, each of the multiplicity of inbound queues storing a plurality of packets that are to be separately transmitted.**

The Examiner has taken the position that claims 1 and 20 are obvious over Erimli in view of Barucchi. However, it appears from the Examiner's rejections that the characterization of the Erimli reference is incorrect. In fact, the Examiner continues to assert that "Erimli teaches an outbound queue...capable of storing a plurality of inbound queues." However, Applicant respectfully submits that Erimli merely discloses a prior art system in which a single frame is enqueued in accordance with standard prior art systems as described in the background section of Applicant's specification. More specifically, Erimli discloses a queueing structure and method for prioritization of frames in a network switch. See Title. A multiport switch enables communication of data packets between network stations. See Col. 4, lines 21-24. The switch has a number of ports, each of which as a receive first in-first out (FIFO) buffer and transmit FIFO. See col. 6, lines 1-16. Data packets are received and stored in the corresponding receive FIFO. See col. 6, lines 10-16. The frame pointer and associated information is placed into the appropriate output queue of the transmit port. See col. 22, lines 37-54. Specifically, the port vector FIFO places the frame pointer into the top of the appropriate output queue. See col. 8, lines 16-22. The output queues provide temporary storage for frame pointers when they are queued for transmission. Queueing takes the form of the port vector FIFO writing frame pointers into the various output queues. See col. 10, lines 46-56. Thus, the port output queues hold entries for frames to be forwarded to the 100 Mb/s ports. See col. 11, lines 20-23. A frame pointer points to the memory location in the external memory where the frame is stored. See col. 8, lines 1-3. While Erimli discloses that the output queue structure is not limited to frame pointers as entries (see col. 9, lines 28-37), Erimli neither discloses nor suggests queueing queues (or queue pointers) as entries.

Erimli neither discloses nor suggests an outbound queue that is capable of storing or otherwise identifying a plurality of inbound queues. Rather, as the Examiner indicates, Erimli discloses a prior art outbound queue that stores a plurality of packets, rather than storing or otherwise identifying a plurality of inbound queues. Similarly, Erimli neither discloses nor suggests transferring one of the plurality of inbound queues (storing a plurality of packets) to such an outbound queue (or an entry in an outbound queue), and therefore neither discloses nor suggests determining when one of a plurality of inbound queues is ready to be moved to an outbound queue. Rather, as desribed above, Erimli merely discloses enqueueing a single frame and therefore a single entry in the inbound queue onto an outbound queue, and therefore teaches away from transferring or enqueueing an entire queue of packets in an outbound queue. In addition, Erimli neither discloses nor suggests providing a plurality of inbound queues for an inbound port and classifying a packet in one of the plurality of inbound queues. In fact, a single inbound queue and transmit queue is disclosed for each port, as described above. See col. 6, lines 1-16. Barucchi fails to cure the deficiencies of the primary reference. As such, combining the cited references would fail to achieve the desired result (reducing CPU overhead in the forwarding process).

Erimli does disclose the handling of frames stored in an inbound queue. However, it is important to note that a frame is transmitted between network points as a unit complete with addressing and necessary protocol control information (similarly to a packet), and are therefore separately transmittable. Erimli neither discloses nor suggests transferring a queue of packets to an entry in an outbound queue. Rather, Erimli discloses transferring individual frame pointers to the outbound queue, as described above. Thus, a single frame or single entry from the inbound queue is enqueued in (i.e., transferred to) a single entry in an outbound queue, rather than enqueueing the entire inbound queue storing a plurality of packets which are separately transmittable in an entry in an outbound queue. In other words, the cited art teaches that a single transmittable message (e.g., frame) is transferred to an outbound queue at a time (to an entry in the outbound queue), rather than transferring a plurality of separately transmittable messages (e.g., frames) at a time (to an entry in the outbound queue). Stated another way, the presently claimed invention transfers more than one separately transmittable message (e.g., packet) to an entry in the outbound queue, and is therefore superior to the prior art, which transfers only one separately transmittable message (e.g., frame) to an entry in an outbound queue. It is clear that in Erimli it is frames that are separately stored in entries in the outbound message queue, rather than an entire inbound queue (e.g., plurality of separately transmissible messages) that is transferred and stored in an entry in the outbound queue. As such, Erimli teaches away from the claimed invention. Barucchi does disclose classifying and storing an inbound packet in a queue as the

Examiner asserts. However, Barucchi fails to cure the deficiencies of the primary reference as set forth above. Accordingly, Applicant respectfully traverses the Examiner's assertion.

None of the references, separately or in combination, disclose or suggest the problems present in the prior art or the solutions presented by the presently claimed invention. The cited references neither disclose nor suggest the problem or limitations of systems in which per-packet processing is performed, or otherwise teach a solution in which CPU overhead is reduced by limiting the per-packet processing performed by a router. In fact, the combination of the cited references would fail to achieve the desired result, which is to reduce CPU overhead in the forwarding process by reducing per-packet processing. In other words, the combination of the cited references would merely achieve a standard prior art system in which packets (or frames) are transferred individually to an outbound queue. Based on the foregoing, it is submitted that the independent claims are patentable over the cited references. In addition, it is submitted that the dependent claims are also patentable for at least the same reasons. In addition, the dependent claims recite additional limitations. For example, dependent claims 2 and 22 further recite asserting an interrupt when it is determined that one of the plurality of inbound queues is ready to be moved to an outbound queue. Dependent claim 3 recites sorting an inbound packet into one of a plurality of inbound queues.

As another example, claim 4 recites "the selected one of the plurality of inbound queues corresponding to one of a plurality of outbound queues" and "transferring the selected one of the plurality of inbound queues storing a plurality of packets to the outbound queue associated with the outbound port such that a reference to the selected inbound queue storing a plurality of packets is stored in a single one of a plurality of entries in the outbound queue." In this manner, an inbound queue storing a plurality of packets is transferred to an entry in the outbound queue and therefore "enqueued" in an outbound queue, thereby reducing per-packet processing and therefore reducing CPU overhead in the forwarding process. In the Examiner's response to Applicant's arguments, the Examiner asserts that "Applicant argues prior art does not teach claim limitation, specifically, "enqueueing an entire queue of packets in an outbound queue". In response to the above-mentioned argument, it is noted that the features upon which applicant relies are not recited in the rejected claim(s)." As can be seen from the limitations recited in claim 4, for example, it is submitted that this Examiner's interpretation of the claims is incorrect.

Various dependent claims recite techniques for memory management. For instance, dependent claims 5 and 23 recite obtaining an available packet buffer from a free pool of available packet buffers in which a packet may then be stored.

Various criteria may be used to determine when one of the plurality of inbound queues is ready to be moved to an outbound queue. For instance, this determination may include determining whether the number of packets in an inbound queue exceeds a maximum number of packets as recited in claims 6 and 24, determining whether the number of bytes in the inbound queue exceeds a maximum number of bytes as recited in claims 7 and 25, when a free pool of available memory has been depleted as recited in claims 8 and 26, or when a maximum time limit has been exceeded as recited in claims 9 and 27. None of the recited references, separately or in combination, discloses or suggests limiting per-packet processing by processing a queue of packets when it is determined that the inbound queue is ready to be moved to an outbound queue.

Since claims 2-9 and claims 21-27 each depend from independent claim 1 or 20, it is respectfully submitted that they are also patentable over the cited references. The additional limitations recited in the independent claims or the dependent claims are not further discussed as the above-discussed limitations are clearly sufficient to distinguish the claimed invention from the cited references. Accordingly, it is respectfully submitted that the outstanding rejections of claims 1-9 and 20-27 under 35 USC §103(a) are improper and should be reversed.

(b) The combination of Erimli, Barucchi and Clark neither discloses nor suggests the invention of claims 45-48, 52

The Examiner has rejected claims 45-48, 52 under 35 U.S.C. §103(a) as being unpatentable over U.S. Patent No. 6,487,212 (Erimli) in view of U.S. Patent No. 5,392,401 (Barucchi), and further in view of Clark. Applicant respectfully asserts that since claims 45-48 and 52 depend from independent claim 1, claims 45-48 and 52 are allowable for at least the reasons set forth above in section (a). In addition, claims 45-48 recite additional limitations. For instance, claim 45, which depends from claim 2, recites transferring the one of the plurality of inbound queues to the outbound queue or an outbound controller associated with the-outbound queue when the interrupt is asserted. In other words, an inbound queue may be transferred to an outbound queue or an outbound controller by a CPU, as recited in claim 46. A variety of mechanisms may be used to enqueue an inbound queue in an outbound queue. Specifically, as recited in claim 47, a reference to the inbound queue may be transferred to an entry in the outbound queue. Similarly, as recited in claim 48, a pointer to the inbound queue may be transferred to an entry in the outbound queue. Thus, as recited in claim 52, the outbound queue includes a plurality of entries, each of the plurality of entries storing or identifying one of the multiplicity of inbound queues. As set forth above, Erimli and Barucchi, separately or in combination, fail to disclose or suggest transferring an entire inbound queue to an entry in an outbound queue. Clark fails to cure the deficiencies of Erimli and Barucchi. In fact, it appears that the Examiner cites Clark solely for its disclosure of prior art encryption mechanisms. Since none of claims 45-48 or 52 relate to encryption and Clark does not appear to disclose features that are material to the rejections of claims 45-48 or 52, Applicant respectfully submits that the outstanding rejection of claims 45-48 and 52 under 35 USC §103(a) is improper and should be reversed.

(c) The combination of Erimli and Barucchi neither discloses nor suggests the invention of claims 10-18, 28-36

The Examiner has rejected claims 10-18 and 28-36 under 35 U.S.C. §103(a) as being unpatentable over U.S. Patent No. 6,487,212 (Erimli) in view of U.S. Patent No. 5,392,401 (Barucchi). Independent claims 10 and 28 relate to an **outbound controller** or outbound controller method wherein the outbound controller transfers an inbound queue storing a plurality of packets to one of a plurality of entries in an outbound queue. In this manner, per-packet processing is substantially reduced. More specifically, pending claim 10 recites:

10. A method for providing an outbound controller for a router, the router having an inbound port and an outbound port, a memory, and a CPU, the outbound controller being adapted for forwarding packets at the outbound port, the method comprising:

providing an **outbound queue** associated with the outbound port and **being capable of storing a plurality of inbound queues**;

receiving a notification to handle an inbound queue, the inbound queue storing a plurality of packets that are to be separately transmitted;

transferring the inbound queue storing a plurality of packets to a single entry in the outbound queue associated with the outbound port such that a reference to the inbound queue storing a plurality of packets is stored in one of a plurality of entries in the outbound queue; and

repeating the receiving and transferring steps for the plurality of inbound queues such that a reference to each of the plurality of inbound queues is separately stored in a different one of the plurality of entries in the outbound queue. (Emphasis added).

Pending claim 28 recites:

28. An outbound controller for a router, the router having an inbound port and an outbound port, a memory, and a CPU, the outbound controller being adapted for forwarding packets at the outbound port, comprising:

a module adapted for **receiving a notification to handle an inbound queue associated with the inbound port, the inbound queue storing a plurality of packets**;

wherein at least one of the CPU and the memory are adapted for storing an **outbound queue associated with the outbound port, the outbound queue being capable of storing a plurality of inbound queues in a plurality of entries in the outbound queue, each of the**

plurality of inbound queues storing a plurality of packets that are to be separately transmitted; and

a queue transferring module adapted for **transferring the inbound queue storing a plurality of packets to a single entry in the outbound queue such that a reference to the inbound queue is stored in the entry in the outbound queue**.

The Examiner has taken the position that claims 10-18 and 28-36 are obvious over Erimli in view of Barucchi. As set forth above, the Examiner continues to assert that "Erimli teaches an outbound queue...capable of storing a plurality of inbound queues." However, Applicant respectfully asserts that the Examiner's characterization of the Erimli reference is largely inaccurate, as set forth above in section (a). In addition, in the Examiner's response to Applicant's arguments, the Examiner asserts that "Applicant argues prior art does not teach claim limitation, specifically, "enqueueing an entire queue of packets in an outbound queue". In response to the above-mentioned argument, it is noted that the features upon which applicant relies are not recited in the rejected claim(s)." As can be seen from the limitations recited in claims 10 and 28, which both recite **transferring an inbound queue storing a plurality of packets to a single entry in an outbound queue associated with an outbound port**, it is submitted that this Examiner's interpretation of the claims is incorrect.

With respect to claim 10, Applicant respectfully submits that neither Erimli nor Barucchi, separately or in combination, discloses or suggests a method for providing an outbound controller for a router, the router having an inbound port and an outbound port, a memory, and a CPU, the outbound controller being adapted for forwarding packets at the outbound port, where an **outbound queue** is associated with the outbound port and **being capable of storing a plurality of inbound queues, receiving a notification to handle an inbound queue, the inbound queue storing a plurality of packets that are to be separately transmitted, transferring the inbound queue storing a plurality of packets to a single entry in the outbound queue associated with the outbound port such that a reference to the inbound queue storing a**

plurality of packets is stored in one of a plurality of entries in the outbound queue, or **repeating the receiving and transferring steps for the plurality of inbound queues such that a reference to each of the plurality of inbound queues is separately stored in a different one of the plurality of entries in the outbound queue**. In fact, as set forth above in section (a), Erimli teaches away from transferring a queue of packets to a single entry in an outbound queue. As a result, the combination of the cited references would fail to achieve the desired reduction in CPU overhead in the forwarding process. Accordingly, it is respectfully submitted that the outstanding rejection of claim 10 under 35 U.S.C. §103(a) is improper and should be reversed.

With respect to claim 28, Applicant respectfully submits that neither Erimli nor Barucchi, separately or in combination, discloses or suggests an outbound controller for a router, the router having an inbound port and an outbound port, a memory, and a CPU, the outbound controller being adapted for forwarding packets at the outbound port, which includes a module adapted for **receiving a notification to handle an inbound queue associated with the inbound port, the inbound queue storing a plurality of packets**, wherein at least one of the CPU and the memory are adapted for storing an outbound queue associated with the outbound port, the **outbound queue being capable of storing a plurality of inbound queues in a plurality of entries in the outbound queue, each of the plurality of inbound queues storing a plurality of packets that are to be separately transmitted**, or a queue transferring module adapted for **transferring the inbound queue storing a plurality of packets to a single entry in the outbound queue such that a reference to the inbound queue is stored in the entry in the outbound queue**. In fact, as set forth above, Erimli teaches away from transferring a queue of packets to a single entry in an outbound queue. As a result, the combination of the cited references would fail to achieve the desired reduction in CPU overhead in the forwarding process. Accordingly, it is respectfully submitted that the outstanding rejection of claim 28 under 35 U.S.C. §103(a) is improper and should be reversed.

Applicant respectfully asserts that since claims 11-18 and 29-36 depend from independent claim 10 or 18, it is respectfully submitted that they also are not in any way obvious over the cited references for at least these reasons. In addition, the dependent claims recite additional limitations that are neither disclosed nor suggested by the cited references. For instance, claims 11 and 29 recite receiving a notification from the CPU to handle the inbound queue. Claims 12 and 30 recite transmitting packets stored in the outbound queue, while claims 13 and 31 recites selectively discarding packets stored in the outbound queue. Claims 14 and 32 explicitly recites the process of transmitting packets stored in an outbound queue, which includes obtaining a next one of the plurality of inbound queues stored in the outbound queue, transmitting selected packets stored in the inbound queue, and releasing memory associated with the inbound queue. For instance, releasing memory may include storing the released memory in a free pool of available packet buffers as recited in claims 15 and 33, forming a new inbound queue to be used by an inbound controller as recited in claims 16 and 34, or forming a queue to be used by the outbound controller during bi-directional operation as recited in claims 17 and 35. Moreover, an inbound queue may be transferred according to the priority of the inbound queue, as recited in claims 18 and 36. Accordingly, it is respectfully submitted that the outstanding rejections of claims 11-18 and 29-36 under 35 U.S.C. §103(a) are improper and should be reversed.

(d) The combination of Erimli, Barucchi and Clark neither discloses nor suggests the invention of claim 51

The Examiner has rejected claim 51 under 35 U.S.C. §103(a) as being unpatentable over U.S. Patent No. 6,487,212 (Erimli) in view of U.S. Patent No. 5,392,401 (Barucchi), and further in view of Clark. This rejection is respectfully traversed. Applicant respectfully asserts that since claim 51 depends from independent claim 10, claim 51 is allowable for at least the reasons set forth above in section (c). In addition, claim 51 recites additional limitations. Specifically, claim 51, which depends from claim 10, recites wherein transferring the inbound queue to a single entry in the outbound queue is performed by the CPU in response to an interrupt. Clark fails to cure the deficiencies of Erimli and Barucchi. In fact, it appears that the Examiner cites Clark solely for its disclosure of prior art encryption mechanisms. Since claim 51 does not relate to encryption and Clark does not appear to disclose features that are material to the rejections of claim 51, Applicant respectfully submits that the outstanding rejection of claim 51 under35 USC §103(a) is improper and should be reversed.

(e) The combination of Erimli and Barucchi neither discloses nor suggests the invention of claim 19

The Examiner has rejected claim 19 under 35 U.S.C. §103(a) as being unpatentable over U.S. Patent No. 6,487,212 (Erimli) in view of U.S. Patent No. 5,392,401 (Barucchi). Independent claim 19 relates to a **router** method in which a packet that is received is classified in one of a plurality of inbound queues associated with an inbound port and wherein one of the plurality of inbound queues storing a plurality of packets is transferred to one of a plurality of outbound queues. In this manner, per-packet processing is substantially reduced. More specifically, pending claim 19 recites:

19. A method for forwarding a packet in a router, the router having a plurality of inbound ports and a plurality of outbound ports, a memory, and a CPU, the method comprising:

providing a plurality of inbound queues for one of the plurality of inbound ports;

providing a **plurality of outbound queues, each one of the plurality of outbound queues corresponding to one of the plurality of outbound ports and being capable of storing a plurality of inbound queues**;

receiving an inbound packet at the one of the plurality of inbound ports;

classifying the inbound packet in a selected one of the plurality of inbound queues according to packet sorting criteria;

storing the inbound packet in the selected one of the plurality of inbound queues;

repeating the steps of receiving, classifying, and storing until an interrupt is asserted; and

transferring one of the plurality of inbound queues storing a plurality of packets to one of the plurality of outbound queues corresponding to the packet sorting criteria when the interrupt is asserted such that a reference to the one of the plurality of inbound queues storing a plurality of packets is stored in a single one of a plurality of entries in the one of the plurality of outbound queues, wherein each of the plurality of packets in the one of the plurality of inbound queues is to be separately transmitted. (Emphasis added).

The Examiner has taken the position that claim 19 is obvious over Erimli in view of Barucchi. As set forth above, the Examiner continues to assert that "Erimli teaches an outbound queue...capable of storing a plurality of inbound queues." However, Applicant respectfully asserts that the Examiner's characterization of the Erimli reference is largely inaccurate, as set forth above in section (a). In addition, in the Examiner's response to Applicant's arguments, the Examiner asserts that "Applicant argues prior art does not teach claim limitation, specifically, "enqueueing an entire queue of packets in an outbound queue". In response to the above-mentioned argument, it is noted that the features upon which applicant relies are not recited in the rejected claim(s)." As can be seen from the limitations recited in claim 19, which recites **transferring one of the plurality of inbound queues storing a plurality of packets to one of the plurality of outbound queues corresponding to the packet sorting criteria when the interrupt is asserted such that a reference to the one of the plurality of inbound queues storing a plurality of packets is stored in a single one of a plurality of entries in the one of the plurality of outbound queues, wherein each of the plurality of packets in the one of the plurality of inbound queues is to be separately transmitted**, it is submitted that this Examiner's interpretation of the claims is incorrect.

With respect to claim 19, Applicant respectfully submits that neither Erimli nor Barucchi, separately or in combination, discloses or suggests a method for forwarding a packet in a router, the router having a plurality of inbound ports and a plurality of outbound ports, a memory, and a CPU, in which a plurality of inbound queues are provided for one of the plurality of inbound ports and a **plurality of outbound queues are provided, each one of the plurality of outbound queues corresponding to one of the plurality of outbound ports and being capable of storing a plurality of inbound queues**, where the method includes receiving an inbound packet at the one of the plurality of inbound ports, classifying the inbound packet in a selected one of the plurality of inbound queues according to packet sorting criteria, and storing the inbound packet in the selected one of the plurality of inbound queues, where **the steps of receiving, classifying, and storing are repeated until an interrupt is asserted**. Moreover, as set forth above, neither of the cited references, separately or in combination, discloses or suggests **transferring one of the plurality of inbound queues storing a plurality of packets to one of the plurality of outbound queues corresponding to the packet sorting criteria when the interrupt is asserted such that a reference to the one of the plurality of inbound queues storing a plurality of packets is stored in a single one of a plurality of entries in the one of the plurality of outbound queues, wherein each of the plurality of packets in the one of the plurality of inbound queues is to be separately transmitted**. In fact, as set forth above, Erimli teaches away from transferring a queue of packets to a single entry in an outbound queue. As a result, the combination of the cited references would fail to achieve the desired reduction in CPU overhead in the forwarding process. Accordingly, it is respectfully submitted that the outstanding rejection of claim 19 under 35 U.S.C. §103(a) is improper and should be reversed.

(f) The combination of Erimli, Barucchi and Clark neither discloses nor suggests the invention of claims 53 and 54.

The Examiner has rejected claims 53 and 54 under 35 U.S.C. §103(a) as being unpatentable over U.S. Patent No. 6,487,212 (Erimli) in view of U.S. Patent No. 5,392,401 (Barucchi), and further in view of Clark. Applicant respectfully asserts that since claims 53 and 54 are merely computer-readable medium and apparatus claims, respectively, corresponding to claim 19, discussed above in section (e), and are allowable for the reasons set forth above in section (e). It is unclear why the rejection of claims 53 and 54 differs from that of claim 19. Regardless, Clark fails to cure the deficiencies of Erimli and Barucchi. In fact, it appears that the Examiner cites Clark solely for its disclosure of prior art encryption mechanisms. Since claims 53 and 54 do not relate to encryption and Clark does not appear to disclose features that are material to the rejections of claims 53 and 54, Applicant respectfully submits that the outstanding rejections of claims 53 and 54 under35 USC §103(a) are improper as set forth above and should be reversed.

(g) The combination of Erimli and Barucchi neither discloses nor suggests the invention of claims 37-42.

The Examiner has rejected claims 37-42 under 35 U.S.C. §103(a) as being unpatentable over U.S. Patent No. 6,487,212 (Erimli) in view of U.S. Patent No. 5,392,401 (Barucchi). Independent claim 37 relates to a **router** in which a plurality of inbound queues are associated with one of a plurality of inbound ports, a plurality of outbound queues are associated with a plurality of outbound ports, and wherein each of the outbound queues is capable of storing a plurality of inbound queues such that a reference to each of the plurality of inbound queues is stored in a different one of a plurality of entries in the outbound queue. In this manner, per-packet processing is substantially reduced. More specifically, pending claim 37 recites:

37. A router having a plurality of inbound ports and a plurality of outbound ports, a memory, and a CPU, comprising:

an inbound controller coupled to one of the plurality of inbound ports, the inbound controller being adapted for receiving an inbound packet;

wherein the memory has stored therein:

a **plurality of inbound queues for the one of the plurality of inbound ports**, each one of the plurality of inbound queues being **capable of storing a plurality of packets that are to be separately transmitted**;

a plurality of outbound queues, each one of the plurality of outbound queues corresponding to one of the plurality of outbound ports and being capable of storing a plurality of inbound queues such that a reference to each of the plurality of inbound queues is stored in a different one of a plurality of entries in the one of the plurality of outbound queues; and

a **classifier coupled to the inbound controller, the classifier being adapted for classifying the inbound packet in a selected one of the plurality of inbound queues according to packet sorting criteria, the selected one of the plurality of inbound queues being associated with one of the plurality of outbound queues**;

wherein the **inbound controller is adapted for storing the inbound packet in the selected one of the plurality of inbound queues**.

The Examiner has taken the position that claims 37-42 are obvious over Erimli in view of Barucchi. As set forth above, the Examiner continues to assert that "Erimli teaches an outbound queue...capable of storing a plurality of inbound queues." However, Applicant respectfully asserts that the Examiner's characterization of the Erimli reference is largely inaccurate, as set forth above in section (a). Applicant respectfully submits that the cited references, separately or in combination, neither discloses nor suggests a router including a **plurality of inbound queues for the one of the plurality of inbound ports**, each one of the plurality of inbound queues being **capable of storing a plurality of packets that are to be separately transmitted, a plurality of outbound queues, each one of the plurality of outbound queues corresponding to one of the**

plurality of outbound ports and being capable of storing a plurality of inbound queues such that a reference to each of the plurality of inbound queues is stored in a different one of a plurality of entries in the one of the plurality of outbound queues, and a **classifier coupled to the inbound controller, the classifier being adapted for classifying the inbound packet in a selected one of the plurality of inbound queues according to packet sorting criteria, the selected one of the plurality of inbound queues being associated with one of the plurality of outbound queues**, wherein the **inbound controller is adapted for storing the inbound packet in the selected one of the plurality of inbound queues**. Moreover, as set forth above, Erimli teaches away from storing a plurality of inbound queues in an outbound queue by storing a reference to each of the inbound queues in a different one of a plurality of entries in an outbound queue. The cited references neither disclose nor suggest the problem or limitations of systems in which per-packet processing is performed, or otherwise teach a solution in which CPU overhead is reduced by limiting the per-packet processing performed by a router. Accordingly, it is respectfully submitted that the outstanding rejection of claim 37 under 35 U.S.C. §103(a) is improper and should be reversed.

Applicant respectfully asserts that since claims 38-42 depend from independent claim 37, it is respectfully submitted that they also are not in any way obvious over the cited references for at least these reasons. For instance, with respect to claim 38, the router further includes an outbound controller coupled to the inbound controller, wherein the inbound controller **selects one of the plurality of inbound queues to be transferred to the outbound controller**, and wherein the outbound controller is adapted for **storing a reference to the selected one of the plurality of inbound queues in an entry in one of the plurality of outbound queues associated with the packet sorting criteria** and transmitting packets stored in the one of the plurality of outbound queues.

In addition, claims 39-42 recite various memory management aspects which are neither disclosed nor suggested by the cited art. For example, claim 39 recites an inbound controller adapted for obtaining memory for an inbound packet, claim 40 recites an outbound controller adapted for releasing selected packet buffers, while claim 41 which depends from claim 40 recites releasing the selected packet buffers into a free pool of available packet buffers. With respect to claim 42, which depends from claim 38, the outbound controller includes a memory releasing module adapted for **providing a new inbound queue to the inbound controller to replace the selected one of the plurality of inbound queues**. As set forth above, Erimli teaches away from managing (e.g., transferring) queues of packets, and therefore Erimli teaches away from providing a new inbound queue to replace an inbound queue selected to be transferred to an outbound queue. Accordingly, it is respectfully submitted that the outstanding rejections of claims 37-42 under 35 U.S.C. §103(a) are improper and should be reversed.

(h) The combination of Erimli, Barucchi and Clark neither discloses nor suggests the invention of claims 43, 49-50

The Examiner has rejected claims 43 and 49-50 under 35 U.S.C. §103(a) as being unpatentable over Erimli over Barucchi in view of Clark. Independent claim 43 relates to an **encryption system** in which an encryption box is adapted for encrypting one of a plurality of inbound queues received by an outbound controller and wherein the outbound controller includes an outbound classifier adapted for classifying the encrypted inbound queue in one of a plurality of outbound queues and storing a reference to the encrypted inbound queue in a single entry in the one of the plurality of outbound queues. In this manner, per-packet processing is substantially reduced. More specifically, pending claim 43 recites:

43. An encryption system, comprising:

an inbound controller adapted for receiving an inbound packet;

a classifier coupled to the inbound controller and adapted for classifying and storing the inbound packet in one of a plurality of inbound queues;

an outbound controller adapted for **receiving the one of the plurality of inbound queues, the one of the plurality of inbound queues storing a plurality of packets to be separately transmitted**; and

an encryption box coupled to the outbound controller, the encryption box being adapted for **encrypting the one of the plurality of inbound queues to provide an encrypted inbound queue to the outbound controller for transmission, wherein the outbound controller includes an outbound classifier adapted for classifying the encrypted inbound queue in one of a plurality of outbound queues associated with a plurality of outbound ports, the outbound controller being adapted for storing a reference to the encrypted inbound queue in a single entry in the one of the plurality of outbound queues, and transmitting data stored in the one of the plurality of outbound queues**.

Applicant respectfully asserts that, for at least the reasons set forth above, the combination of the cited references neither discloses nor suggests the claimed invention. None of the cited references, separately or in combination, enable CPU overhead to be reduced in the forwarding process through the handling of queues of packets such that an inbound queue of packets is transferred to a single entry in an outbound queue (by storing a reference to the inbound queue). In fact, as set forth above, Erimli teaches away from such a system.

In addition, it is important to note that in claim 43, the inbound queue for which a reference (e.g., pointer) is stored in the outbound queue is encrypted prior to storing the reference in the entry of the outbound queue. The Examiner asserts that "Clark discusses as prior art encryption mechanisms, which encrypt an entire buffer (queue or set of data packets)" citing column 1, lines 34 through column 2, line 10. While Clark does disclose the encryption of a "buffer," Applicant respectfully submits that the Examiner has mischaracterized the reference by interpreting the term "buffer" to mean a queue or set of data packets. In fact, in no manner does Clark disclose or suggest the encryption of an inbound queue **storing a plurality of packets to**

be separately transmitted. Even if Clark were interpreted to teach or suggest the encryption of a queue of packets, Clark fails to cure the deficiencies of the primary references, which fail to teach or suggest transferring a queue of packets or otherwise reducing CPU overhead by limiting the per-packet processing performed by a router. Accordingly, it is respectfully submitted that the outstanding rejection of claim 43 under 35 U.S.C. §103(a) is improper and should be reversed.

Applicant respectfully asserts that since claims 49-50 depend from independent claim 43, it is respectfully submitted that they also are not in any way obvious over the cited references for at least these reasons. Moreover, the dependent claims recite additional limitations that are neither disclosed nor suggested by the cited art. For instance, claim 49 recites an encryption box that does not encrypt each of the packets. In other words, the encryption box encrypts the inbound queue as a whole. Alternatively, claim 50 recites an encryption box adapted for encrypting an inbound queue as an entity such that a single encryption step is performed. In this manner, an inbound queue may be encrypted using a variety of encryption schemes prior to transferring the inbound queue to an entry in the outbound queue. As set forth above, none of the cited references, separately or in combination, discloses or suggests encryption of an inbound queue prior to transferring the encrypted inbound queue to an entry in the outbound queue. Accordingly, it is respectfully submitted that the outstanding rejections of claims 49-50 under 35 U.S.C. §103(a) are improper and should be reversed.

(i) Conclusion

In view of the forgoing, it is respectfully submitted that none of the pending claims are anticipated or rendered obvious by any reasonable combination of the art of record. Accordingly, the pending rejection of all of the claims under 35 USC §103 should be reversed.

Respectfully Submitted,
BEYER WEAVER & THOMAS, LLP

Elise R. Heilbrunn
Reg. No. 42,649

Beyer Weaver & Thomas, LLP
P.O. Box 778
Berkeley, California
(510) 843-6200

IX. APPENDIX

CLAIMS ON APPEAL

1. (Previously Amended) A method for providing an inbound controller for a router, the router having an inbound port and an outbound port, a memory, and a CPU, the inbound controller being adapted for receiving an inbound packet at the inbound port, the method comprising:

providing a plurality of inbound queues for the inbound port;

receiving an inbound packet at the inbound port;

classifying the inbound packet in a selected one of the plurality of inbound queues according to packet sorting criteria;

storing the inbound packet in the selected one of the plurality of inbound queues; and

determining when one of the plurality of inbound queues storing a plurality of packets is ready to be moved to an entry in an outbound queue associated with the outbound port, the outbound queue being capable of storing a reference to a multiplicity of inbound queues such that a reference to each of the multiplicity of inbound queues is separately stored in a different one of a plurality of entries in the outbound queue, each of the multiplicity of inbound queues storing a plurality of packets to be separately transmitted.

2. (Previously Amended) The method as recited in claim 1, further including:

asserting an interrupt when it is determined that one of the plurality of inbound queues is ready to be moved to an outbound queue.

3. (Original Claim) The method as recited in claim 1, wherein classifying the inbound packet includes:

selecting inbound packet sorting criteria;

obtaining packet sorting data for the inbound packet, the packet sorting data being associated with the packet sorting criteria; and

sorting the inbound packet into one of the plurality of inbound queues according to the packet sorting data.

4. (Previously Amended) The method as recited in claim 1, the selected one of the plurality of inbound queues corresponding to one of a plurality of outbound queues, the method further comprising:

transferring the selected one of the plurality of inbound queues storing a plurality of packets to the outbound queue associated with the outbound port such that a reference to the selected inbound queue storing a plurality of packets is stored in a single one of a plurality of entries in the outbound queue.

5. (Original Claim) The method as recited in claim 1, wherein storing the inbound packet includes:

obtaining an available packet buffer from a free pool of available packet buffers;

placing the inbound packet in the packet buffer; and

storing the packet buffer in the inbound queue.

6. (Original Claim) The method as recited in claim 1, wherein determining when one of the plurality of inbound queues is ready to be moved to an outbound queue includes:

determining whether a number of packets in one of the plurality of inbound queues exceeds a maximum number of packets.

7. (Original Claim) The method as recited in claim 1, wherein determining when one of the plurality of inbound queues is ready to be moved to an outbound queue includes:

determining whether a number of bytes in one of the plurality of inbound queues exceeds a maximum number of bytes.

8. (Original Claim) The method as recited in claim 1, wherein determining when one of the plurality of inbound queues is ready to be moved to an outbound queue further includes:

determining whether a free pool of available memory has been depleted.

9. (Original Claim) The method as recited in claim 1, wherein determining when one of the plurality of inbound queues is ready to be moved to an outbound queue further includes:

determining whether a maximum time limit has been exceeded.

10. (Previously Amended) A method for providing an outbound controller for a router, the router having an inbound port and an outbound port, a memory, and a CPU, the outbound controller being adapted for forwarding packets at the outbound port, the method comprising:

providing an outbound queue associated with the outbound port and being capable of storing a plurality of inbound queues;

receiving a notification to handle an inbound queue, the inbound queue storing a plurality of packets that are to be separately transmitted;

transferring the inbound queue storing a plurality of packets to a single entry in the outbound queue associated with the outbound port such that a reference to the inbound queue storing a plurality of packets is stored in one of a plurality of entries in the outbound queue; and

repeating the receiving and transferring steps for the plurality of inbound queues such that a reference to each of the plurality of inbound queues is separately stored in a different one of the plurality of entries in the outbound queue.

11. (Original Claim) The method as recited in claim 10, wherein receiving the notification includes:

receiving a notification from the CPU to handle the inbound queue.

12. (Original Claim) The method as recited in claim 10, further including:

transmitting packets stored in the outbound queue.

13. (Original Claim) The method as recited in claim 10, wherein transmitting packets includes:

selectively discarding packets stored in the outbound queue.

14. (Original Claim) The method as recited in claim 10, wherein transmitting packets stored in the outbound queue further includes:

obtaining a next one of the plurality of inbound queues stored in the outbound queue;

transmitting selected packets stored in the next one of the plurality of inbound queues; and

releasing memory associated with the next one of the plurality of inbound queues.

15. (Original Claim) The method as recited in claim 14, wherein releasing the memory includes:

storing the released memory in a free pool of available packet buffers.

16. (Original Claim) The method as recited in claim 14, wherein releasing the memory includes:

forming a new inbound queue to be used by an inbound controller.

17. (Original Claim) The method as recited in claim 14, wherein releasing the memory includes:

forming a queue to be used by the outbound controller during bi-directional operation.

18. (Previously Amended) The method as recited in claim 10, wherein transferring the inbound queue to the outbound queue further includes:

ascertaining a priority of the inbound queue; and

transferring the inbound queue to a single entry in the outbound queue according to the priority of the inbound queue.

19. (Previously Amended) A method for forwarding a packet in a router, the router having a plurality of inbound ports and a plurality of outbound ports, a memory, and a CPU, the method comprising:

providing a plurality of inbound queues for one of the plurality of inbound ports;

providing a plurality of outbound queues, each one of the plurality of outbound queues corresponding to one of the plurality of outbound ports and being capable of storing a plurality of inbound queues;

receiving an inbound packet at the one of the plurality of inbound ports;

classifying the inbound packet in a selected one of the plurality of inbound queues according to packet sorting criteria;

storing the inbound packet in the selected one of the plurality of inbound queues;

repeating the steps of receiving, classifying, and storing until an interrupt is asserted; and

transferring one of the plurality of inbound queues storing a plurality of packets to one of the plurality of outbound queues corresponding to the packet sorting criteria when the interrupt is asserted such that a reference to the one of the plurality of inbound queues storing a plurality of packets is stored in a single one of a plurality of entries in the one of the plurality of outbound queues, wherein each of the plurality of packets in the one of the plurality of inbound queues is to be separately transmitted.

20. (Previously Amended) An inbound controller for a router, the router having an inbound port and an outbound port, a memory, and a CPU, the inbound controller being adapted for receiving an inbound packet at the inbound port, comprising:

a packet receiving module coupled to the inbound port, the packet receiving module being adapted for receiving an inbound packet;

wherein the memory has stored therein:

a plurality of inbound queues for the inbound port;

a classifier adapted for classifying the inbound packet in a selected one of the plurality of inbound queues according to packet sorting criteria;

a packet storing module coupled to the classifier, the packet storing module being adapted for storing the inbound packet in the selected one of the plurality of inbound queues; and

a module adapted for determining when one of the plurality of inbound queues is ready to be moved to an entry in an outbound queue associated with the outbound port, the outbound queue being capable of storing a multiplicity of inbound queues, a reference to each of the multiplicity of inbound queues being stored in a different one of a plurality of entries in the outbound queue, each of the multiplicity of inbound queues storing a plurality of packets that are to be separately transmitted.

21. (Original Claim) The inbound controller as recited in claim 20, further including:

a module adapted for providing the determined one of the plurality of inbound queues.

22. (Original Claim) The inbound controller as recited in claim 20, further including:

a module adapted for asserting an interrupt when it is determined that one of the plurality of inbound queues is ready to be moved by the CPU to the outbound queue.

23. (Original Claim) The inbound controller as recited in claim 20, wherein the packet storing module includes:

a memory obtaining module adapted for obtaining an available packet buffer from a free pool of available packet buffers;

a module adapted for placing the inbound packet in the packet buffer; and

a module adapted for storing the packet buffer in the inbound queue.

24. (Original Claim) The inbound controller as recited in claim 20, wherein the module adapted for determining when one of the plurality of inbound queues is ready to be moved to an outbound queue includes:

a module adapted for determining whether a number of packets in one of the plurality of inbound queues exceeds a maximum number of packets.

25. (Original Claim) The inbound controller as recited in claim 20, wherein the module adapted for determining when one of the plurality of inbound queues is ready to be moved to an outbound queue includes:

a module adapted for determining whether a number of bytes in one of the plurality of inbound queues exceeds a maximum number of bytes.

26. (Original Claim) The inbound controller as recited in claim 20, wherein the module adapted for determining when one of the plurality of inbound queues is ready to be moved to an outbound queue includes:

a module adapted for determining whether a free pool of available memory has been depleted.

27. (Original Claim) The inbound controller as recited in claim 20, wherein the module adapted for determining when one of the plurality of inbound queues is ready to be moved to an outbound queue includes:

a module adapted for determining whether a maximum time limit has been exceeded.

28. (Previously Amended) An outbound controller for a router, the router having an inbound port and an outbound port, a memory, and a CPU, the outbound controller being adapted for forwarding packets at the outbound port, comprising:

a module adapted for receiving a notification to handle an inbound queue associated with the inbound port, the inbound queue storing a plurality of packets;

wherein at least one of the CPU and the memory are adapted for storing an outbound queue associated with the outbound port, the outbound queue being capable of storing a plurality of inbound queues in a plurality of entries in the outbound queue, each of the plurality of inbound queues storing a plurality of packets that are to be separately transmitted; and a queue transferring module adapted for transferring the inbound queue storing a plurality of packets to a single entry in the outbound queue such that a reference to the inbound queue is stored in the entry in the outbound queue.

29. (Original Claim) The outbound controller as recited in claim 28, wherein the module adapted for receiving the notification includes a module adapted for receiving the notification from the CPU.

30. (Original Claim) The outbound controller as recited in claim 28, further including:

a module adapted for transmitting packets stored in the outbound queue.

31. (Original Claim) The outbound controller as recited in claim 30, wherein the module adapted for transmitting packets includes:

a module adapted for selectively discarding packets stored in the outbound queue.

32. (Original Claim) The outbound controller as recited in claim 30, wherein the module adapted for transmitting packets stored in the outbound queue includes:

a module adapted for obtaining a next one of the plurality of inbound queues stored in the outbound queue;

a packet transmission module adapted for transmitting selected packets stored in the next one of the plurality of inbound queues; and

a memory releasing module adapted for releasing memory associated with the next one of the plurality of inbound queues.

33. (Original Claim) The outbound controller as recited in claim 32, wherein the memory releasing module includes:

a module adapted for storing the released memory in a free pool of available packet buffers.

34. (Original Claim) The outbound controller as recited in claim 32, wherein the released memory forms a new inbound queue to be used by an inbound controller.

35. (Original Claim) The outbound controller as recited in claim 32, wherein the released memory forms a queue to be used by the outbound controller during bi-directional operation.

36. (Previously Amended) The outbound controller as recited in claim 28, wherein the queue transferring module is adapted for transferring the inbound queue to a single entry in the outbound queue according to a priority of the inbound queue.

37. (Previously Amended) A router having a plurality of inbound ports and a plurality of outbound ports, a memory, and a CPU, comprising:

an inbound controller coupled to one of the plurality of inbound ports, the inbound controller being adapted for receiving an inbound packet;

wherein the memory has stored therein:

a plurality of inbound queues for the one of the plurality of inbound ports, each one of the plurality of inbound queues being capable of storing a plurality of packets that are to be separately transmitted;

a plurality of outbound queues, each one of the plurality of outbound queues corresponding to one of the plurality of outbound ports and being capable of storing a plurality of inbound queues such that a reference to each of the plurality of inbound queues is stored in a different one of a plurality of entries in the one of the plurality of outbound queues; and

a classifier coupled to the inbound controller, the classifier being adapted for classifying the inbound packet in a selected one of the plurality of inbound queues according to packet sorting criteria, the selected one of the plurality of inbound queues being associated with one of the plurality of outbound queues;

wherein the inbound controller is adapted for storing the inbound packet in the selected one of the plurality of inbound queues.

38. (Previously Amended) The router as recited in claim 37, further including:

an outbound controller coupled to the inbound controller;

wherein the inbound controller selects one of the plurality of inbound queues to be transferred to the outbound controller;

wherein the outbound controller is adapted for storing a reference to the selected one of the plurality of inbound queues in an entry in one of the plurality of outbound queues associated with the packet sorting criteria and transmitting packets stored in the one of the plurality of outbound queues.

39. (Original Claim)The router as recited in claim 37, wherein the inbound controller further includes:

a memory obtaining module coupled to the classifier, the memory obtaining module being adapted for obtaining memory for an inbound packet to permit the inbound packet to be stored in the selected one of the plurality of inbound queues in which the inbound packet is classified.

40. (Original Claim) The router as recited in claim 38, wherein the outbound controller further includes:

a memory releasing module adapted for releasing selected packet buffers associated with packets stored in the one of the plurality of outbound queues.

41. (Original Claim) The router as recited in claim 40, wherein the memory further includes a free pool of available packet buffers and the memory releasing module is adapted for releasing the selected packet buffers into the free pool.

42. (Original Claim) The router as recited in claim 38, wherein the outbound controller further includes:

a memory releasing module adapted for providing a new inbound queue to the inbound controller to replace the selected one of the plurality of inbound queues.

43. (Previously Amended) An encryption system, comprising:

an inbound controller adapted for receiving an inbound packet;

a classifier coupled to the inbound controller and adapted for classifying and storing the inbound packet in one of a plurality of inbound queues;

an outbound controller adapted for receiving the one of the plurality of inbound queues, the one of the plurality of inbound queues storing a plurality of packets to be separately transmitted; and

an encryption box coupled to the outbound controller, the encryption box being adapted for encrypting the one of the plurality of inbound queues to provide an encrypted inbound queue to the outbound controller for transmission, wherein the outbound controller includes an outbound classifier adapted for classifying the encrypted inbound queue in one of a plurality of outbound queues associated with a plurality of outbound ports, the outbound controller being adapted for storing a reference to the encrypted inbound queue in a single entry in the one of the plurality of outbound queues, and transmitting data stored in the one of the plurality of outbound queues.

44. (Cancelled)

45. (Previously Amended) The method as recited in claim 2, further comprising:

when the interrupt is asserted, transferring the one of the plurality of inbound queues to an entry in the outbound queue or an outbound controller associated with the outbound queue.

46. (Previously Amended) The method as recited in claim 45, wherein transferring the one of the plurality of inbound queues to an entry in the outbound queue or an outbound controller associated with the outbound queue is performed by the CPU.

47. (Previously Amended) The method as recited in claim 45, wherein transferring the one of the plurality of inbound queues to an entry in the outbound queue or an outbound controller associated with the outbound queue comprises:

transferring a reference to the one of the plurality of inbound queues to an entry in an outbound queue corresponding to a priority associated with the one of the plurality of inbound queues.

48. (Previously Amended) The method as recited in claim 45, wherein transferring the one of the plurality of inbound queues to the outbound queue or an outbound controller associated with the outbound queue comprises:

transferring a pointer to the one of the plurality of inbound queues to an entry in an outbound queue associated with the one of the plurality of inbound queues.

49. (Original Claim) The method as recited in claim 43, wherein the inbound queue stores therein a plurality of packets, and wherein the encryption box does not encrypt each of the plurality of packets.

50. (Original Claim) The method as recited in claim 43, wherein the encryption box is adapted for encrypting the inbound queue as an entity such that a single encryption step is performed.

51. (Previously Amended) The method as recited in claim 10, wherein transferring the inbound queue to a single entry in the outbound queue is performed by the CPU in response to an interrupt.

52. (Currently Amended) The method as recited in claim 1, wherein the outbound queue comprises a plurality of entries, each of the plurality of entries ~~simultaneously~~ storing or identifying one of the multiplicity of inbound queues.

53. (Previously Amended) A computer-readable medium storing thereon computer-readable instructions for forwarding a packet in a router, the router having a plurality of inbound ports and a plurality of outbound ports, a memory, and a CPU, the method comprising:

instructions for providing a plurality of inbound queues for one of the plurality of inbound ports;

instructions for providing a plurality of outbound queues, each one of the plurality of outbound queues corresponding to one of the plurality of outbound ports and being capable of storing a plurality of inbound queues such that a reference to each of the plurality of inbound queues is simultaneously stored in a different one of a plurality of entries in the one of the plurality of outbound queues;

instructions for receiving an inbound packet at the one of the plurality of inbound ports;

instructions for classifying the inbound packet in a selected one of the plurality of inbound queues according to packet sorting criteria;

instructions for storing the inbound packet in the selected one of the plurality of inbound queues;

instructions for repeating the steps of receiving, classifying, and storing until an interrupt is asserted; and

instructions for transferring one of the plurality of inbound queues to one of the plurality of outbound queues corresponding to the packet sorting criteria when the interrupt is asserted such that a pointer to the transferred one of the plurality of inbound queues is stored in one of a plurality of entries in the one of the plurality of outbound queues, the one of the plurality of inbound queues storing a plurality of packets that are to be separately transmitted.

54. (Previously Amended) An apparatus for forwarding a packet in a router, the router having a plurality of inbound ports and a plurality of outbound ports, a memory, and a CPU, the method comprising:

means for providing a plurality of inbound queues for one of the plurality of inbound ports;

means for providing a plurality of outbound queues, each one of the plurality of outbound queues corresponding to one of the plurality of outbound ports and being capable of storing a plurality of inbound queues such that a reference to each of the plurality of inbound queues is stored in a different one of a plurality of entries in the one of the plurality of outbound queues;

means for receiving an inbound packet at the one of the plurality of inbound ports;

means for classifying the inbound packet in a selected one of the plurality of inbound queues according to packet sorting criteria;

means for storing the inbound packet in the selected one of the plurality of inbound queues;

means for repeating the steps of receiving, classifying, and storing until an interrupt is asserted; and

means for transferring one of the plurality of inbound queues to one of the plurality of outbound queues corresponding to the packet sorting criteria when the interrupt is asserted such that a reference to the transferred one of the plurality of inbound queues is stored in a single one of a plurality of entries in the one of the plurality of outbound queues, the one of the plurality of_ inbound queues storing a plurality of packets that are to be separately transmitted.